CameraRental Application

Phase -1 End Project

The application flow :

Consists of the following steps:

Welcome Screen: Display the application name and developer details, along with options for accessing various features.

Camera Listing: Show a list of available cameras for rent, including brand, model, and per-day rental amount.

Rent Camera: Allow the user to select a camera from the listing and specify the rental duration.

Wallet Management: Enable the user to view and manage their wallet balance, including depositing funds.

Close Application: Provide an option to exit the application.

Sprint 1:

User Interface Design and Camera Listing

Welcome screen design

Camera listing screen design

Data structure for storing camera details

Camera listing feature implementation

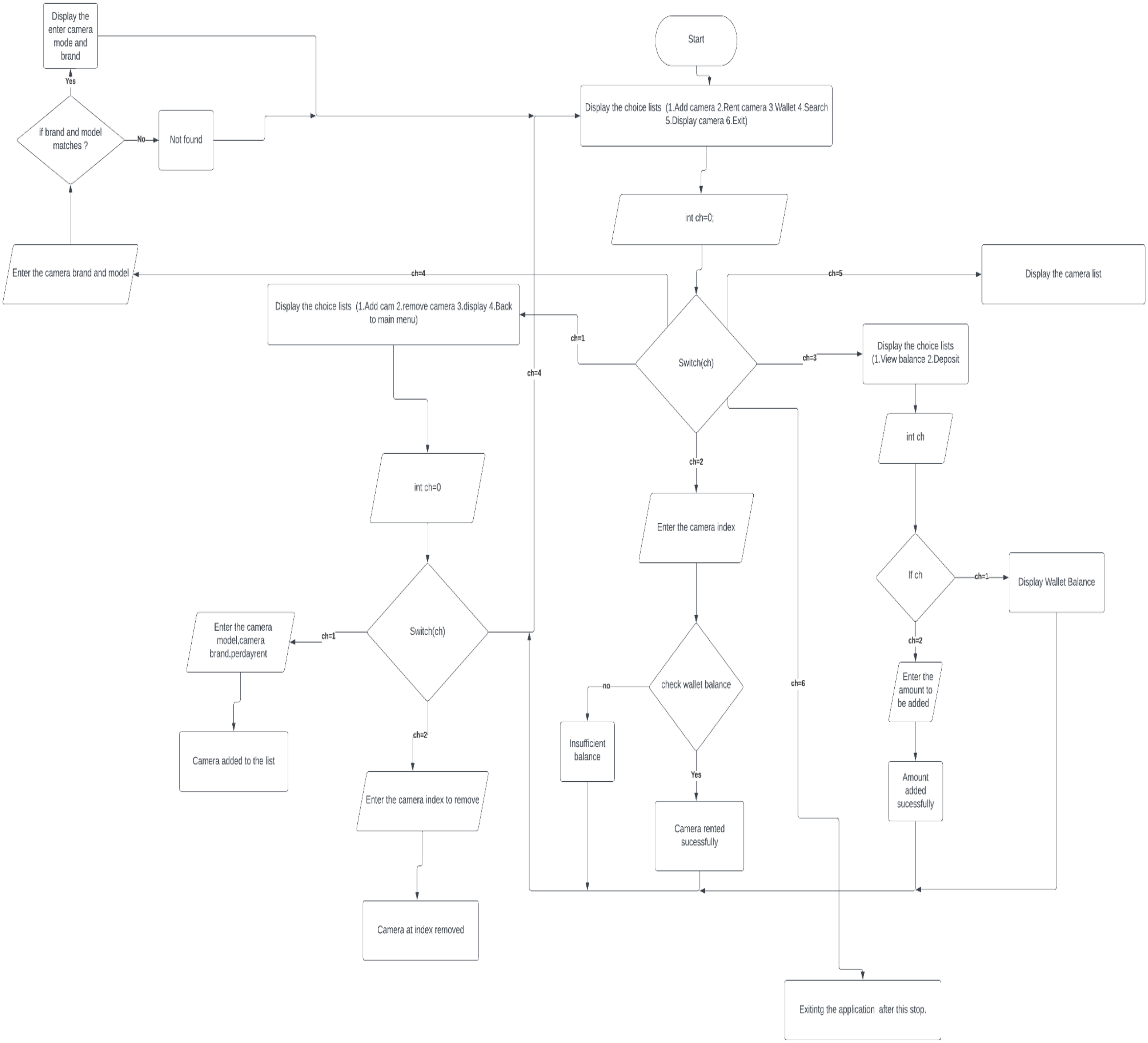
Sprint 2: Camera Selection and Rental

Camera selection screen design

Camera rental screen design

Data structure for storing rental details

Camera rental feature implementation


Start
Display the choice lists (1.Add camera 2.Rent camera 3.Wallet 4.Search 5.Display camera 6.Exit)
int ch=0;
Switch(ch)
ch=1
Display the choice lists (1.Add cam 2.remove camera 3.display 4.Back to main menu)
int ch=0
Switch(ch)
ch=1
Enter the camera model,camera brand.perdayrent
Camera added to the list
ch=2
Enter the camera index to remove
Camera at index removed
ch=4
ch=2
Enter the camera index
check wallet balance
no
Insufficient balance
Yes
Camera rented sucessfully
ch=3
Display the choice lists (1.View balance 2.Deposit
int ch
If ch
ch=1
Display Wallet Balance
ch=2
Enter the amount to be added
Amount added sucessfully
ch=4
Enter the camera brand and model
if brand and model matches ?
Yes
Display the enter camera mode and brand
No
Not found
ch=5
Display the camera list
ch=6
Exitintg the application after this stop.

Sprint 3: Payment

Cart screen design

Checkout screen design

Wallet amount feature implementation

Sprint 4: Testing

Testing and bug fixes

Source Code:

CameraRentalApp.java (consists of main method)

```
package phase1;
import java.util.*;
public class CameraRentalApp {
   public static void main(String[] args) {
       //CameraRentalApplication class is in camera.java file we have created a
instance to call various functions
        CameraRentalApplication app = new CameraRentalApplication();
        //User.java files consists of setting username and password method
        User u=new User();
        Scanner scanner = new Scanner(System.in);
        int choice;

        System.out.println("+---+---+----+----+----+---+----+----+");
        System.out.println("|Welcome To Camera Rental Application|");
        System.out.println("+---+---+---+---+---+---+---+----+---+");
        System.out.println("Please Login to Continue");
        System.out.println("+-------+");
        System.out.print("Username: ");
        System.out.println("\n+-------+");
        String admin=scanner.next();
        u.setName(admin);
        System.out.println("+-------+");
        System.out.print("Password: ");
        System.out.println("\n+-------+");
        String password=scanner.next();
        u.setPassword(password);

 //        System.out.println(u.toString());
 //                 to check whether the enter admin name and password are
 returned.

        //checking whether the entered uname and pwd is crt or not
        if(admin.equalsIgnoreCase(u.getName()) &&
 password.equals(u.getPassword())  ){
        do {
             //displaying the menu screen on every choice
            displayWelcomeScreen();
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    handleAddCamera(app, scanner);
                    break;
                case 2:
                    handleRentCamera(app, scanner);
                    break;
                case 3:
                    handleWalletManagement(app, scanner);
                    break;
                case 4:
                   app.displayCameraList();
                   break;
                case 5:
                   System.out.println("Enter the camera brand:");
                   String brand=scanner.next();
                   System.out.println("Enter the camera model");
                   String model=scanner.next();
                   app.search(brand, model);
                   break;
                case 6:
                    System.out.println("Exiting the application. Goodbye!");
```

```
                     break;
                 default:
                     System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 6);
    }
        else {
            System.out.print("You have entered the Wrong password or username");
        }

}

   private static void displayWelcomeScreen() {
        System.out.println("---+---+----+----+----+---+");
        System.out.println("Camera Rental Application Main Menu");
        System.out.println("---+---+---+---+---+---+---+");
        System.out.println("1. Add a camera");
        System.out.println("2. Rent a camera");
        System.out.println("3. Wallet Management");
        System.out.println("4. Display Camera List");
        System.out.println("5. Want to search any camera?");
        System.out.println("6. Exit");
        System.out.print("Enter your choice: ");
    }
   //method to add camera , inside which called another method from camera.java
   private static void handleAddCamera(CameraRentalApplication app, Scanner
scanner) {

       int choice;

       do {
            System.out.println("---+---+---+---+---+---+---+");
            System.out.println("1. Add a camera");
            System.out.println("2. Remove");
            System.out.println("3. My cameras ");
            System.out.println("4. Back to main menu");
            System.out.print("Enter your choice: ");
              choice=scanner.nextInt();
```

```
           System.out.println("\n---+---+---+---+---+---+---+");
            switch(choice) {
            case 1:
            System.out.println("Add a Camera");
               System.out.println("------------");
               scanner.nextLine(); // Consume newline character
               System.out.print("Enter the brand: ");
               String brand = scanner.nextLine();
               System.out.print("Enter the model: ");
               String model = scanner.nextLine();
               System.out.print("Enter the per-day rental amount: ");
               double rentalAmount = scanner.nextDouble();
               app.addCamera(brand, model, rentalAmount);
               System.out.println("Camera added successfully.");
               break;
            case 2:
                  System.out.println("Enter the index Number to Remove a
camera:");
                  int index=scanner.nextInt();
                  try {
                              app.deleteCamera(index-1);
                        System.out.println("Camera at "+index+"Removed.");
                        } catch (InvalidIndex e) {
                              System.out.println(e.getMessage());}
                  break;
            case 3:
                  app.displayCameraList();
                  break;
            default:
               System.out.println("Enter a Valid choice:");
            }
      }while(choice!=4);
   }
   //to rent a camera and to check if the balance is available or not
   private static void handleRentCamera(CameraRentalApplication app, Scanner
scanner) {
       System.out.println("Rent a Camera");
       System.out.println("-------------");
```

```
        app.displayCameraList();
        if (app.cameraList.isEmpty()) {
            System.out.println("No cameras available for rent.");
            return;
        }
        System.out.print("Enter the index of the camera to rent: ");
        int cameraIndex = scanner.nextInt();
        System.out.print("Enter the rental duration (in days): ");
        int rentalDuration = scanner.nextInt();
        try {
            app.rentCamera(cameraIndex-1, rentalDuration);
        } catch (InsufficientBalanceException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    //wallet prices to show the amount left and also to add the amount
    private static void handleWalletManagement(CameraRentalApplication app,
Scanner scanner) {
        System.out.println("Wallet Management");
        System.out.println("-----------------");
        System.out.println("1. View Wallet Balance");
        System.out.println("2. Deposit Funds");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                app.displayWalletBalance();
                break;
            case 2:
                System.out.print("Enter the amount to deposit: ");
                double amount = scanner.nextDouble();
                app.depositToWallet(amount);
                break;
            default:
                System.out.println("Invalid choice. Returning to the main
menu.");
        }
```

```
   }
}
```

Camera.java(consists of various operations to be performed)

```
package phase1;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;


class Camera {
    private String brand;
    private String model;
    private double perDayRentalAmount;
    private boolean available;

    public Camera( String brand, String model, double perDayRentalAmount) {
        this.brand = brand;
        this.model = model;
        this.perDayRentalAmount = perDayRentalAmount;
        this.available = true;
    }

    // Getters and setters

    public String getBrand() {
        return brand;
    }

    public String getModel() {
        return model;
    }

    public double getPerDayRentalAmount() {
        return perDayRentalAmount;
    }
```

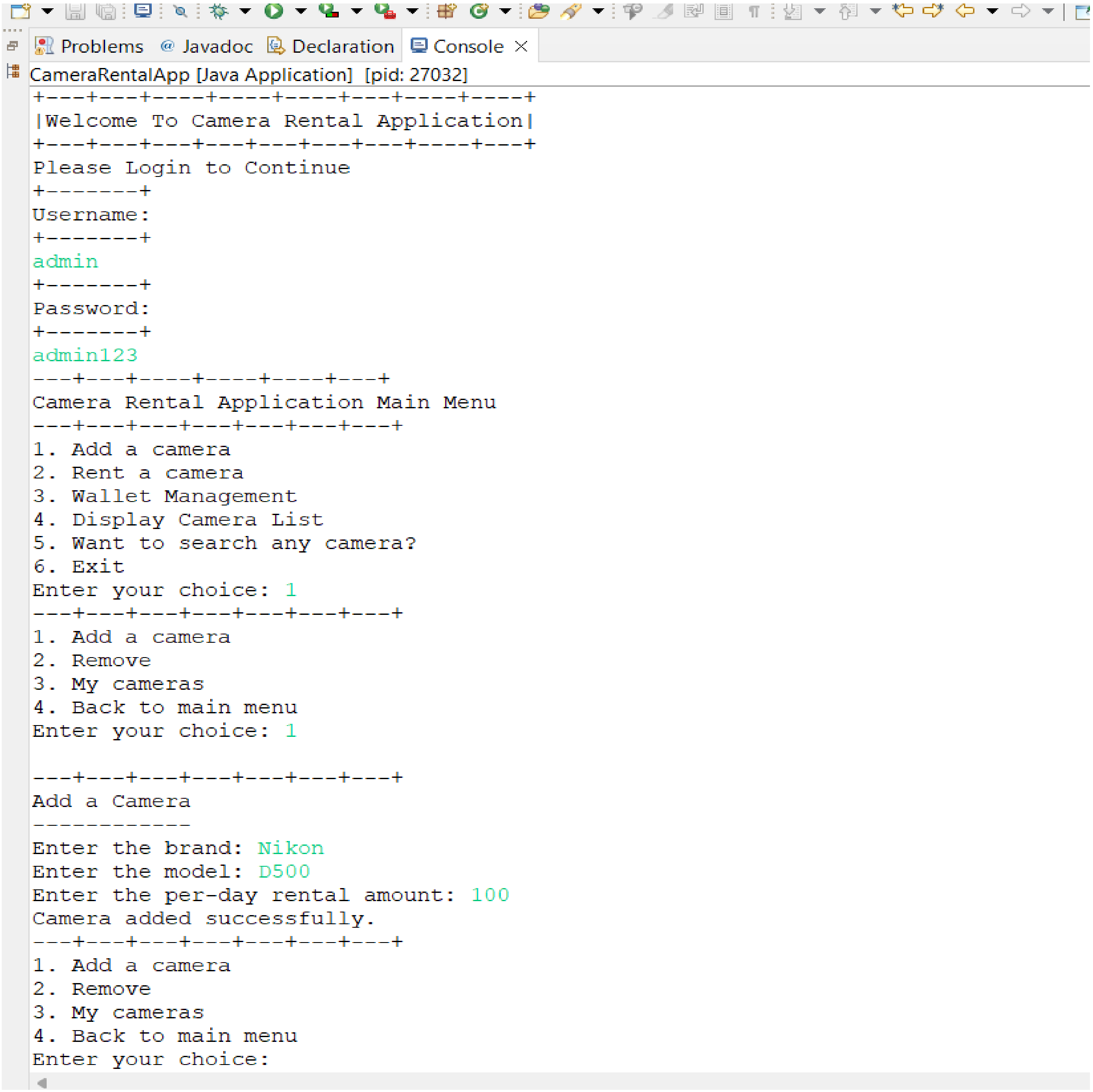
```java
    public boolean isAvailable() {
        return available;
    }

    public void setAvailable(boolean available) {
        this.available = available;
    }
}

//custom exception
class InsufficientBalanceException extends Exception {
      private static final long serialVersionUID = 1L;
      public InsufficientBalanceException(String message) {
        super(message);
    }
}
class InvalidIndex  extends Exception {
      private static final long serialVersionUID = 1L;
      public InvalidIndex(String message) {
            super(message);
      }
}


//this is class  we used many functions such as  append a new camera to the
list and rent a camera and add or deduct money from the wallet.
class CameraRentalApplication {
     List<Camera> cameraList;
    private double walletBalance;

    public CameraRentalApplication() {
        cameraList = new ArrayList<>();
        walletBalance = 0.0;
    }

    public void addCamera(String brand, String model, double
perDayRentalAmount) {
        Camera camera = new Camera(brand, model, perDayRentalAmount);
        cameraList.add(camera);
    }


    public void deleteCamera(int index)  throws InvalidIndex{
      if(index<0) throw new InvalidIndex("Invalid Index");
      if(cameraList.isEmpty()) System.out.println("There are no cameras to 
remove, You can add a camera");
      cameraList.remove(index);
    }
    //displaying the camera's present
    public void displayCameraList() {
        if (cameraList.isEmpty()) {
            System.out.println("No Data Present at This Moment.");
        } else {
            System.out.println("Available Cameras:");
            for (Camera camera : cameraList) {
                if (camera.isAvailable()) {
                    System.out.println("Brand: " + camera.getBrand());
                    System.out.println("Model: " + camera.getModel());
                    System.out.println("Per-day Rental Amount: $" + 
camera.getPerDayRentalAmount());
                    System.out.println("\n--------+-+---------");
                }
            }
        }
    }

    public void rentCamera(int cameraIndex, int rentalDuration) throws 
InsufficientBalanceException {
        Camera camera = cameraList.get(cameraIndex);

        if (!camera.isAvailable()) {
            System.out.println("Camera is not available for rent.");
            return;
        }

        double rentalCost = camera.getPerDayRentalAmount() * rentalDuration;
        if (walletBalance < rentalCost) {
            throw new InsufficientBalanceException("Insufficient balance in the
wallet.");
        }

        walletBalance -= rentalCost;
        camera.setAvailable(false);

        System.out.println("Camera rented successfully.");
    }

    public void displayWalletBalance() {
        System.out.println("Wallet Balance: $" + walletBalance);
    }

    public void depositToWallet(double amount) {
        if (amount <= 0) {
            System.out.println("Invalid deposit amount.");
            return;
        }

        walletBalance += amount;
        System.out.println("Deposit successful.");
    }
    //sorting
    public void sortCameraList(Comparator<Camera> comparator) {
        Collections.sort(cameraList, comparator);
    }
    //searching
      public void search(String model,String brand) {
      if (cameraList.isEmpty()) {
            System.out.println("No Data Present at This Moment.");
        } else {
      for(Camera i :cameraList) {
            if(i.getBrand().equalsIgnoreCase(brand) &&
i.getModel().equalsIgnoreCase(model)) {
                System.out.println("Camera your were looking :");
```

```
                        System.out.println("Brand: " + i.getBrand());
                      System.out.println("Model: " + i.getModel());
                      System.out.println("\n--------+-+---------");
                  }
          }
      }
      }
}
```

User.java

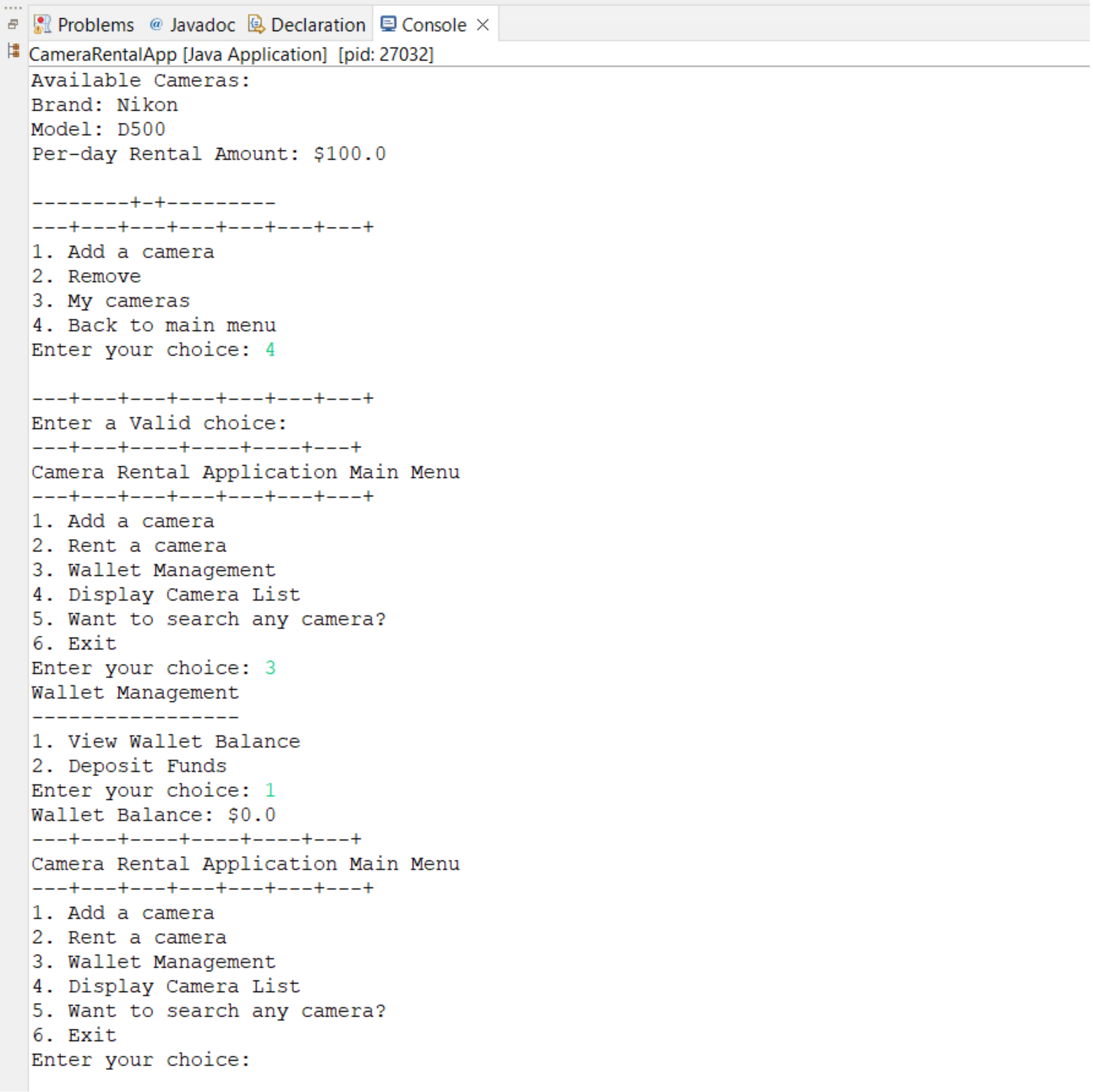
```
package phase1;
public class User {
	String name;
	String password;

	//getters and setters

	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
	@Override
	public String toString() {
		return "User [name=" + name + ", password=" + password + "]";
	}
}
```

Output Screenshots:



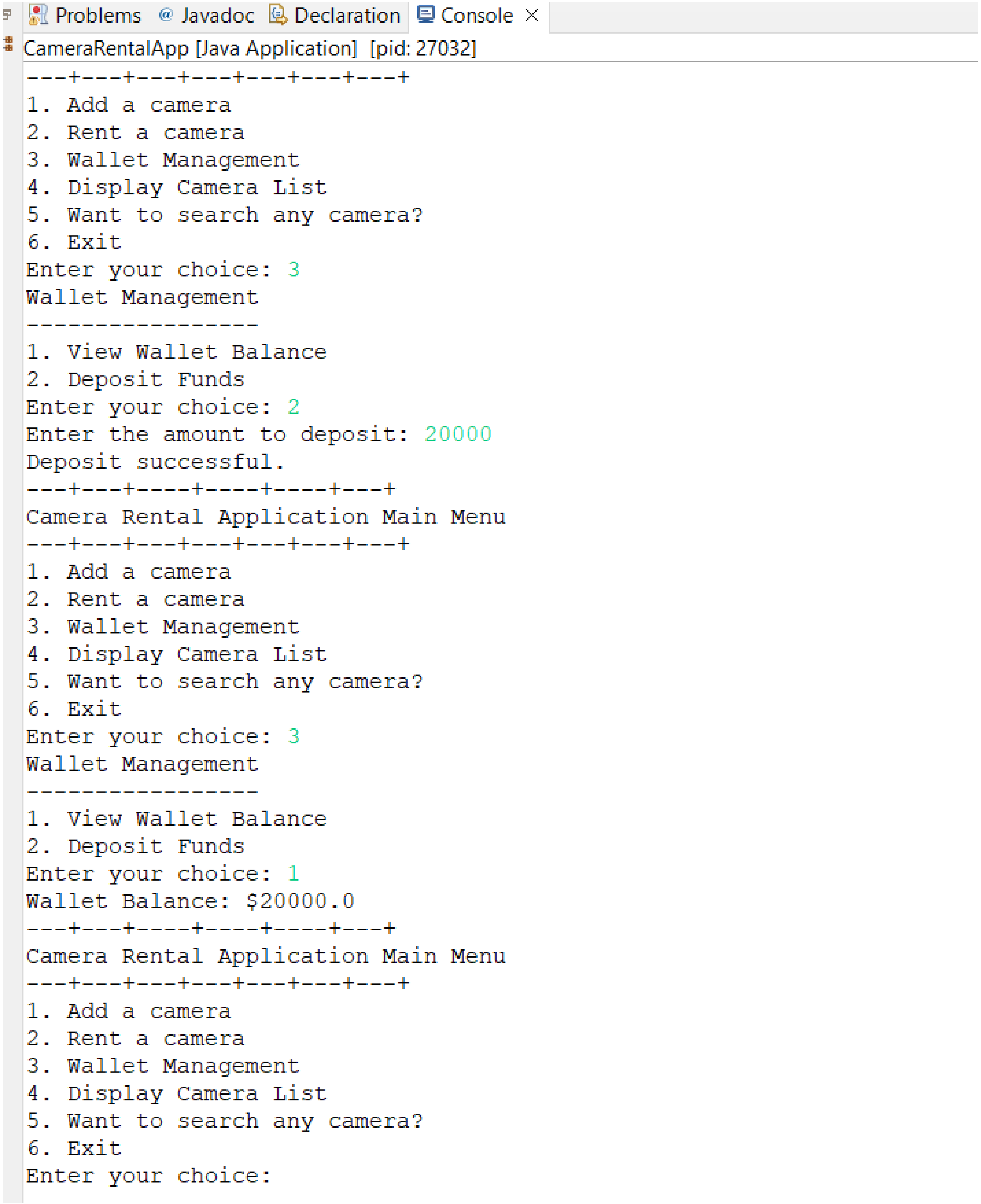
CameraRentalApp [Java Application] [pid: 27032]

```
+---+---+----+----+----+---+----+----+
|Welcome To Camera Rental Application|
+---+---+---+---+---+---+---+----+---+
Please Login to Continue
+--------+
Username:
+--------+
admin
+--------+
Password:
+--------+
admin123
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice: 1
---+---+---+---+---+---+---+
1. Add a camera
2. Remove
3. My cameras
4. Back to main menu
Enter your choice: 1

---+---+---+---+---+---+---+
Add a Camera
------------
Enter the brand: Nikon
Enter the model: D500
Enter the per-day rental amount: 100
Camera added successfully.
---+---+---+---+---+---+---+
1. Add a camera
2. Remove
3. My cameras
4. Back to main menu
Enter your choice:
```

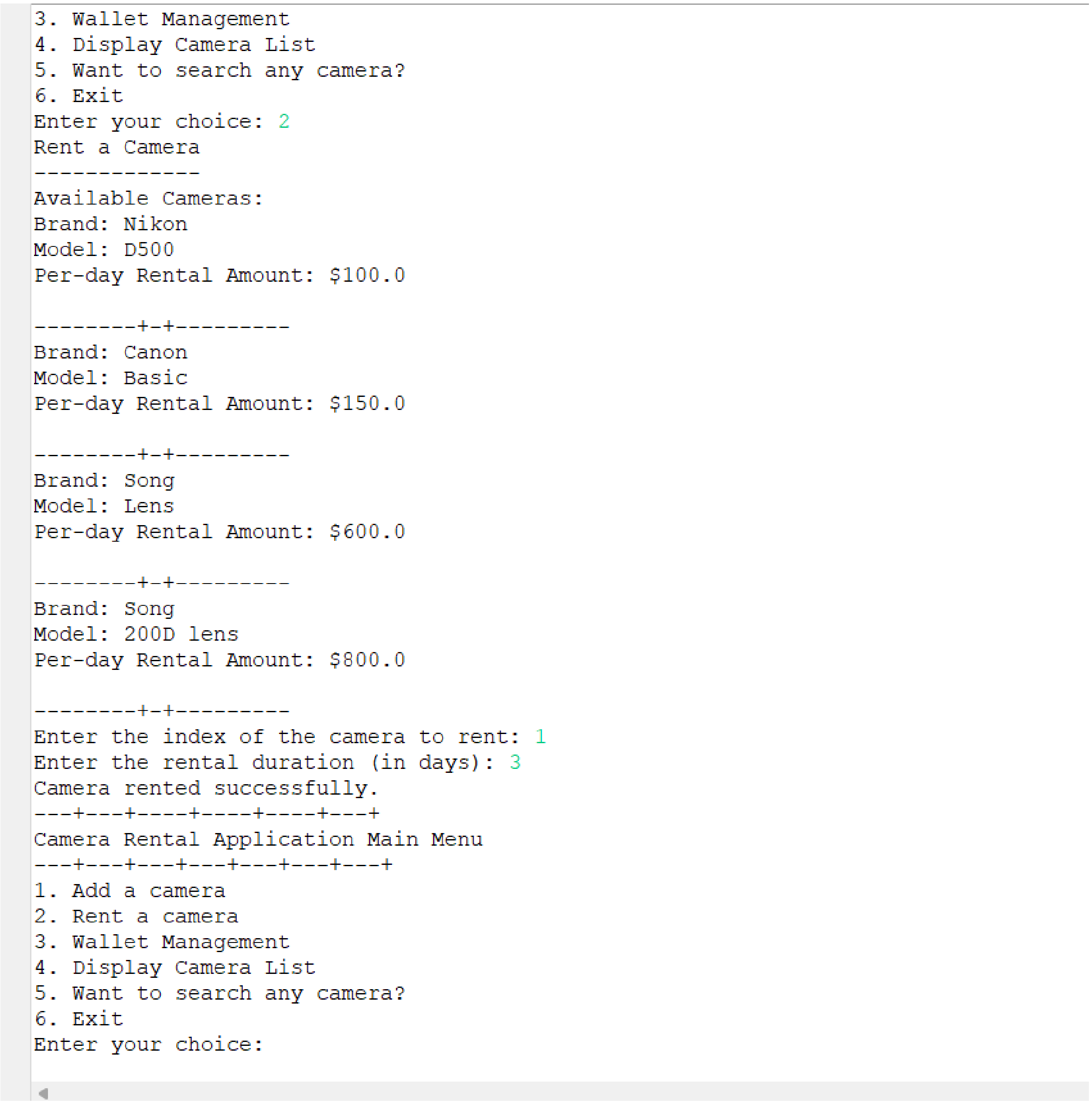


CameraRentalApp [Java Application] [pid: 27032]

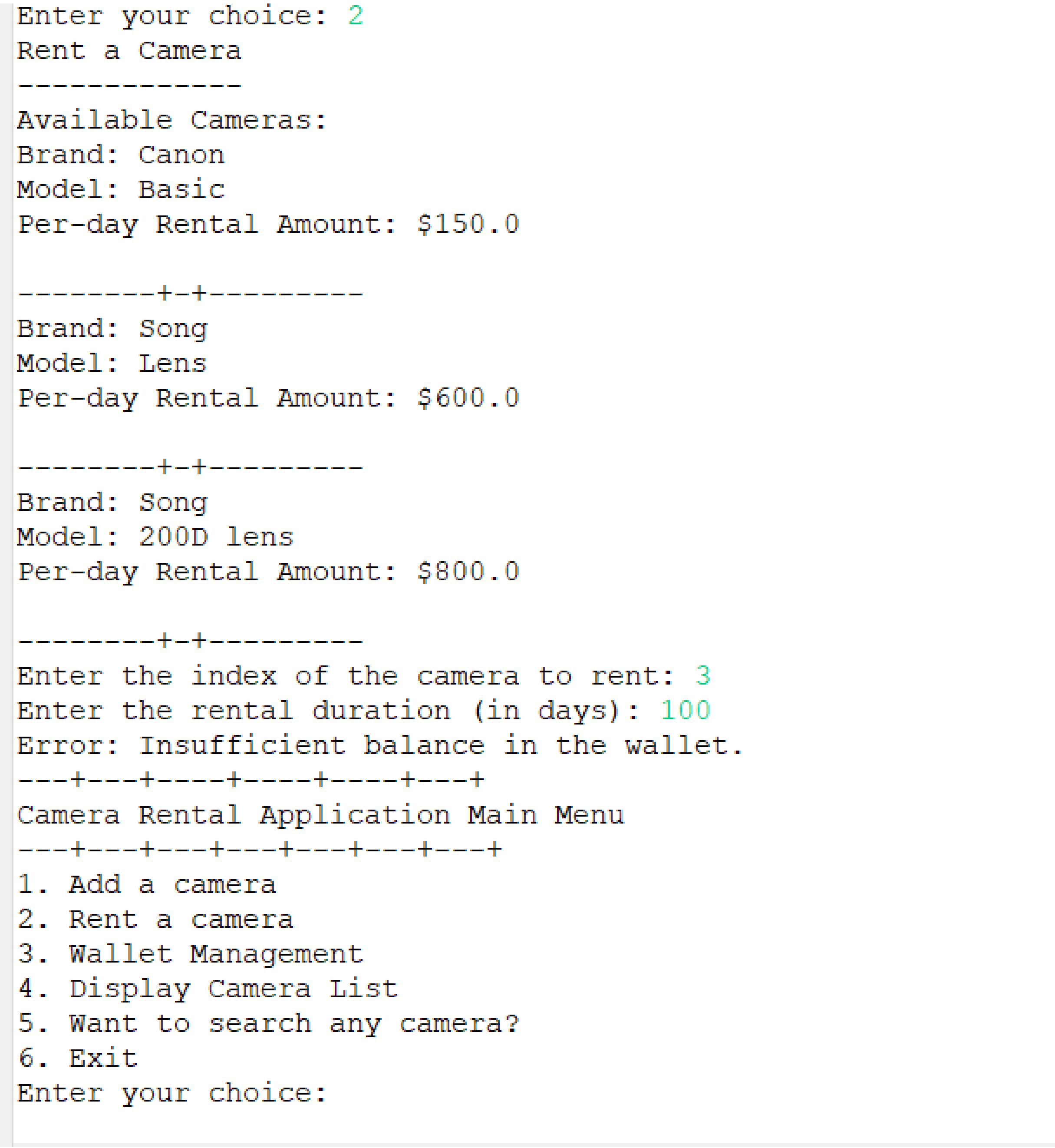
```
Available Cameras:
Brand: Nikon
Model: D500
Per-day Rental Amount: $100.0

--------+-+---------
---+---+---+---+---+---+---+
1. Add a camera
2. Remove
3. My cameras
4. Back to main menu
Enter your choice: 4

---+---+---+---+---+---+---+
Enter a Valid choice:
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice: 3
Wallet Management
-----------------
1. View Wallet Balance
2. Deposit Funds
Enter your choice: 1
Wallet Balance: $0.0
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice:
```



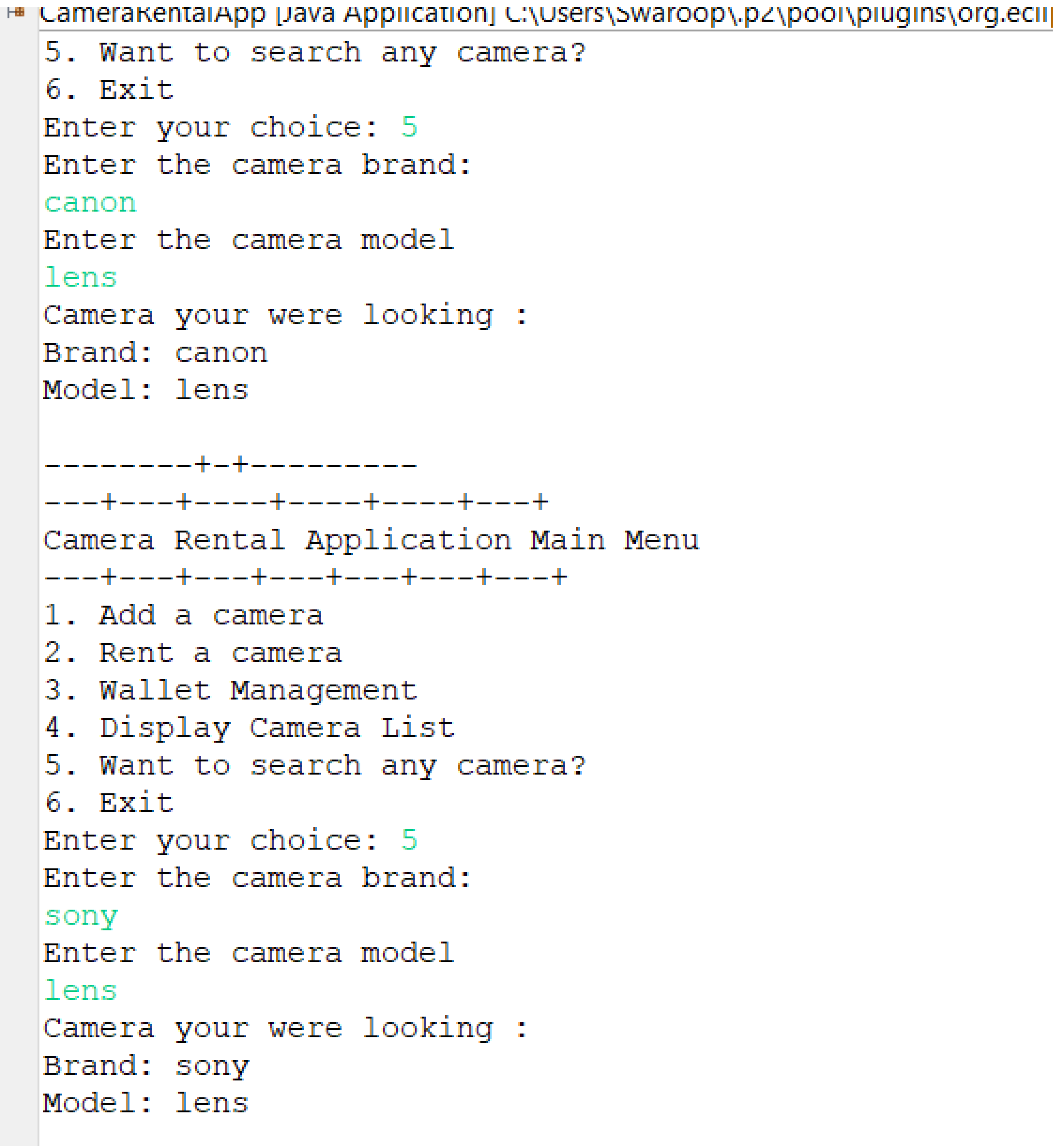
```
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice: 3
Wallet Management
-----------------
1. View Wallet Balance
2. Deposit Funds
Enter your choice: 2
Enter the amount to deposit: 20000
Deposit successful.
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice: 3
Wallet Management
-----------------
1. View Wallet Balance
2. Deposit Funds
Enter your choice: 1
Wallet Balance: $20000.0
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice:
```



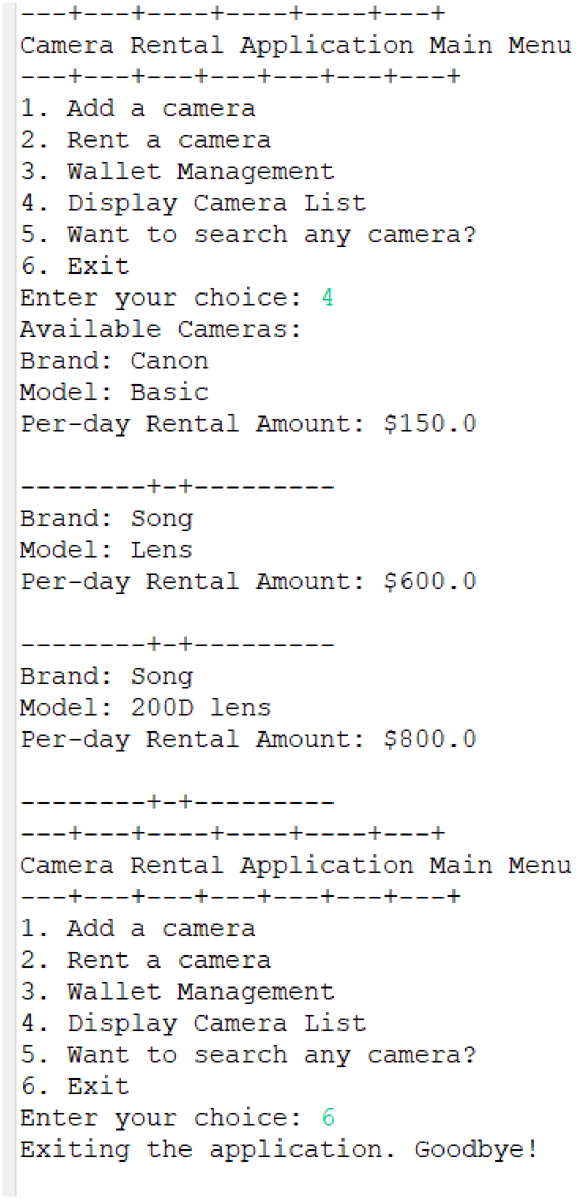
```
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice: 2
Rent a Camera
-------------
Available Cameras:
Brand: Nikon
Model: D500
Per-day Rental Amount: $100.0

--------+-+---------
Brand: Canon
Model: Basic
Per-day Rental Amount: $150.0

--------+-+---------
Brand: Song
Model: Lens
Per-day Rental Amount: $600.0

--------+-+---------
Brand: Song
Model: 200D lens
Per-day Rental Amount: $800.0

--------+-+---------
Enter the index of the camera to rent: 1
Enter the rental duration (in days): 3
Camera rented successfully.
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice:
```



```
Enter your choice: 2
Rent a Camera
-------------
Available Cameras:
Brand: Canon
Model: Basic
Per-day Rental Amount: $150.0

--------+-+---------
Brand: Song
Model: Lens
Per-day Rental Amount: $600.0

--------+-+---------
Brand: Song
Model: 200D lens
Per-day Rental Amount: $800.0

--------+-+---------
Enter the index of the camera to rent: 3
Enter the rental duration (in days): 100
Error: Insufficient balance in the wallet.
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice:
```



```
CameraRentalApp [Java Application] C:\Users\Swaroop\.p2\pool\plugins\org.ecli
5. Want to search any camera?
6. Exit
Enter your choice: 5
Enter the camera brand:
canon
Enter the camera model
lens
Camera your were looking :
Brand: canon
Model: lens

--------+-+---------
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice: 5
Enter the camera brand:
sony
Enter the camera model
lens
Camera your were looking :
Brand: sony
Model: lens
```



```
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice: 4
Available Cameras:
Brand: Canon
Model: Basic
Per-day Rental Amount: $150.0

--------+-+---------
Brand: Song
Model: Lens
Per-day Rental Amount: $600.0

--------+-+---------
Brand: Song
Model: 200D lens
Per-day Rental Amount: $800.0

--------+-+---------
---+---+----+----+----+---+
Camera Rental Application Main Menu
---+---+---+---+---+---+---+
1. Add a camera
2. Rent a camera
3. Wallet Management
4. Display Camera List
5. Want to search any camera?
6. Exit
Enter your choice: 6
Exiting the application. Goodbye!
```

GitCommands:

Git add <file>

Git commit - m “Final assessment”

Git push -u origin master